```
# @title
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable

```
import pandas as pd

df = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	Dev
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	

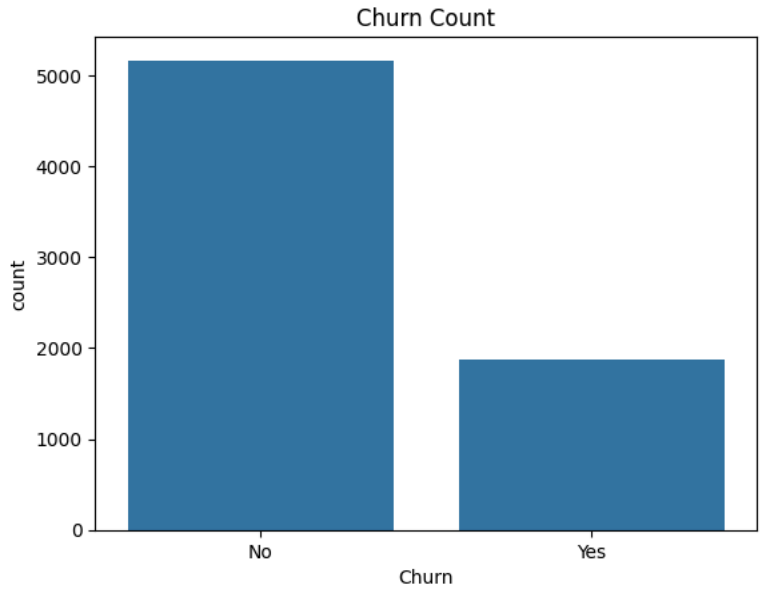
5 rows × 21 columns

```
# Drop missing
df = df.dropna()

# Convert TotalCharges
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df = df.dropna(subset=['TotalCharges'])

# Encode Churn
df['Churn'] = df['Churn'].map({'Yes': 1, 'No': 0})

# Confirm
df.info()
```

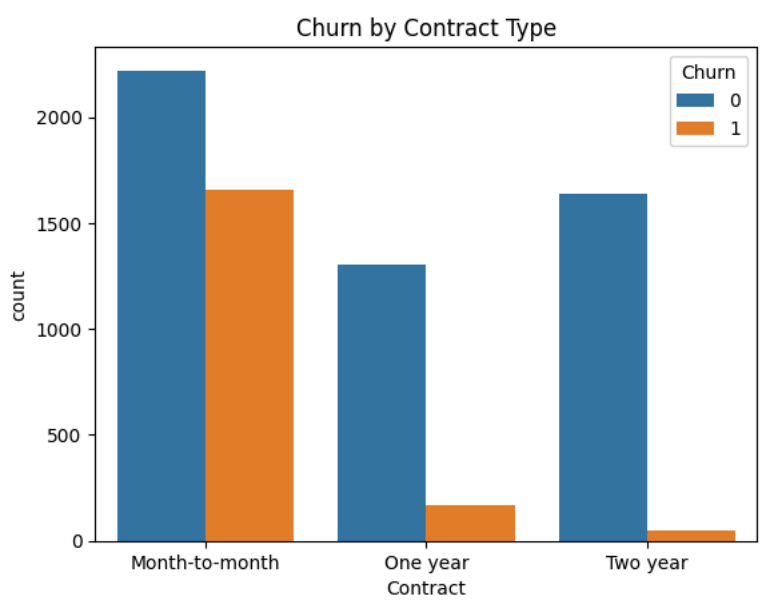
```
<class 'pandas.core.frame.DataFrame'>
Index: 7032 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7032 non-null   object
 1   gender            7032 non-null   object
 2   SeniorCitizen     7032 non-null   int64
 3   Partner           7032 non-null   object
 4   Dependents        7032 non-null   object
 5   tenure            7032 non-null   int64
 6   PhoneService      7032 non-null   object
 7   MultipleLines     7032 non-null   object
 8   InternetService   7032 non-null   object
 9   OnlineSecurity    7032 non-null   object
 10  OnlineBackup      7032 non-null   object
 11  DeviceProtection  7032 non-null   object
 12  TechSupport       7032 non-null   object
 13  StreamingTV       7032 non-null   object
 14  StreamingMovies   7032 non-null   object
 15  Contract          7032 non-null   object
 16  PaperlessBilling  7032 non-null   object
 17  PaymentMethod     7032 non-null   object
 18  MonthlyCharges    7032 non-null   float64
 19  TotalCharges      7032 non-null   float64
 20  Churn             7032 non-null   int64
dtypes: float64(2), int64(3), object(16)
memory usage: 1.2+ MB
```

```
eda_df = df.copy()
```

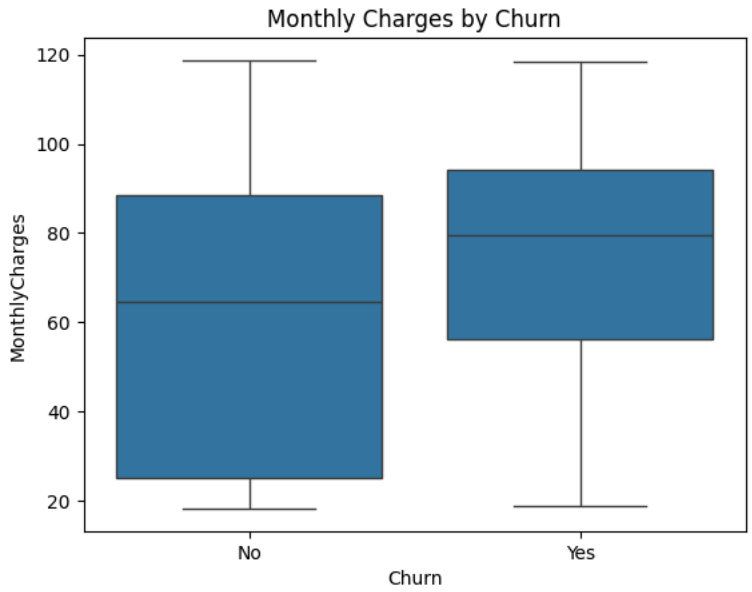
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(x='Churn', data=eda_df)
plt.title("Churn Count")
plt.xticks([0,1], ['No', 'Yes'])
plt.show()
```



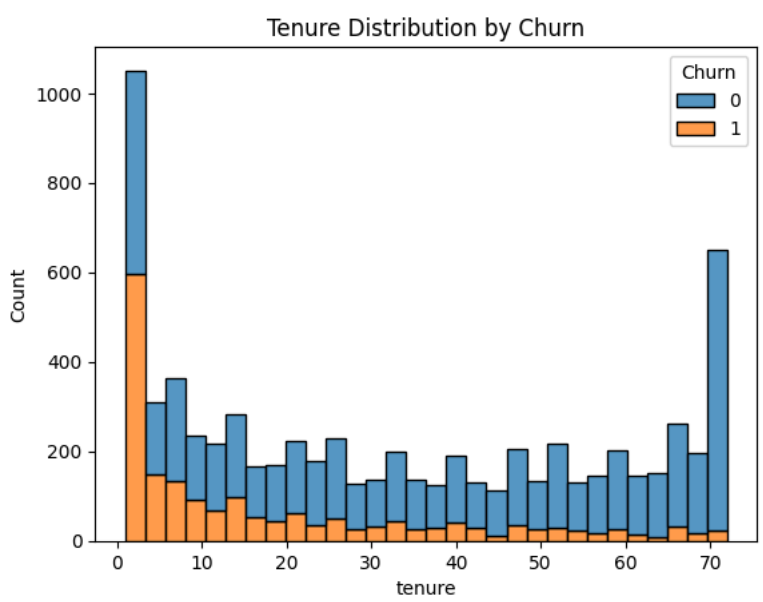

```
sns.countplot(x='Contract', hue='Churn', data=eda_df)
plt.title("Churn by Contract Type")
plt.show()
```




```
sns.boxplot(x='Churn', y='MonthlyCharges', data=eda_df)
plt.title("Monthly Charges by Churn")
plt.xticks([0,1], ['No', 'Yes'])
plt.show()
```




```
sns.histplot(data=eda_df, x='tenure', hue='Churn', multiple='stack', bins=30)
plt.title("Tenure Distribution by Churn")
plt.show()
```




```
# Drop customerID
df = df.drop(['customerID'], axis=1)

# One-hot encoding
df = pd.get_dummies(df, drop_first=True)

df.head()
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	Churn	gender_Male	Partner_Yes	Dependents_Yes	PhoneService_Yes	MultipleLines_ phone servi
0	0	1	29.85	29.85	0	False	True	False	False	T
1	0	34	56.95	1889.50	0	True	False	False	True	Fa
2	0	2	53.85	108.15	1	True	False	False	True	Fa
3	0	45	42.30	1840.75	0	True	False	False	False	T
4	0	2	70.70	151.65	1	False	False	False	True	Fa

5 rows × 31 columns

```
from sklearn.model_selection import train_test_split

X = df.drop('Churn', axis=1)
y = df['Churn']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

# Accuracy
print("Accuracy:", accuracy_score(y_test, y_pred))

# Confusion Matrix
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.7874911158493249
Confusion Matrix:
 [[915 118]
 [181 193]]
/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
import eli5
from eli5.sklearn import PermutationImportance

perm = PermutationImportance(model, random_state=1).fit(X_test, y_test)
eli5.show_weights(perm, feature_names = X_test.columns.tolist())
```

Weight	Feature
0.1137 ± 0.0233	tenure
0.0196 ± 0.0177	TotalCharges
0.0135 ± 0.0032	Contract_Two year
0.0043 ± 0.0129	InternetService_Fiber optic
0.0037 ± 0.0038	PaperlessBilling_Yes
0.0016 ± 0.0030	Partner_Yes
0.0011 ± 0.0096	Contract_One year
0.0010 ± 0.0071	TechSupport_Yes
0.0004 ± 0.0031	gender_Male
0.0003 ± 0.0023	PaymentMethod_Mailed check
0.0001 ± 0.0042	SeniorCitizen
-0.0003 ± 0.0029	StreamingTV_No internet service
-0.0003 ± 0.0014	OnlineBackup_No internet service
-0.0003 ± 0.0019	InternetService_No
-0.0004 ± 0.0042	StreamingTV_Yes
-0.0006 ± 0.0029	MultipleLines_Yes
-0.0007 ± 0.0035	OnlineSecuritv No internet service